

# **Metaheurísticas basadas en Inteligencia Computacional Aplicadas a la Resolución de Problemas de Optimización Restringidos**

Victoria Aragón<sup>†</sup>, Leticia Cagnina<sup>†</sup>, Susana Esquivel<sup>†</sup>

<sup>†</sup>Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)  
Facultad de Ciencias Físicas, Matemáticas y Naturales  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis - Argentina  
Tel: (02652) 420823 / Fax: (02652) 430224  
e-mail: {esquivel, vsaragon, lcagnina}@unsl.edu.ar

## **Resumen**

En este artículo se describen en forma breve algunas de las direcciones de investigación que en la actualidad se están desarrollando dentro de la línea “Optimización Mono y Multiobjetivo” del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC). Uno de los objetivos de esta línea, es el estudio y desarrollo de metaheurísticas aptas para resolver problemas de optimización con restricciones. En particular, el énfasis está puesto en las heurísticas de la inteligencia computacional basados en los paradigmas de inteligencia colectiva y biológicos.

# 1. Introducción

La mayoría de los problemas de optimización del mundo real (ingeniería, diseño de circuitos, corte y empaquetado, etc.) involucran no sólo una función de aptitud, sino que también incluyen restricciones que deben ser satisfechas para que una solución sea *factible*. Luego, el problema de optimización se reduce a encontrar la “mejor” solución factible en el conjunto de soluciones factibles. Problemas con estas características pueden ser formulados, de la manera más general, como problemas de *programación no lineal*.

Un *problema de programación no lineal* se define como:  
Encontrar el vector  $\vec{x}$  el cual optimiza la función:

$$F(\vec{x}) \text{ with } \vec{x} = (x_1, x_2, \dots, x_D) \in \mathcal{F} \subseteq \mathcal{S} \subseteq R^D \quad (1)$$

donde  $F(\vec{x})$  está sujeto a:

$$f_i(\vec{x}) \leq 0 \quad i = 1, 2, \dots, n \quad (2)$$

$$g_e(\vec{x}) = 0 \quad e = 1, 2, \dots, m \quad (3)$$

Existen algoritmos exactos para resolver este tipo de problemas, aunque su aplicabilidad muchas veces está limitada a que tanto la función objetivo como las funciones de restricciones satisfagan ciertas propiedades, a veces muy estrictas (continuidad y diferenciabilidad, entre otras).

En los últimos años distintos métodos heurísticos han sido utilizados para resolver el tipo de problema bajo estudio. Algunas de las heurísticas más empleadas son Tabu Search [23] [24], Ant Colony Optimization [3] [29] [18] [28], y Algoritmos Evolutivos [6] [5] [11].

En años más recientes heurísticas tales como *Particle Swarm* y *Sistemas Inmunes Artificiales* han comenzado a atraer la atención de los investigadores para estudiar su aplicabilidad en el campo de optimización de funciones con restricciones.

## 2. Sistemas Inmunes Artificiales

En años recientes un sistema bio-inspirado ha llamado la atención de los investigadores, el *sistema inmune natural* y su poderosa capacidad de procesamiento de información [22]. El Sistema Inmune (SI) es un sistema muy complejo con varios mecanismos de defensa contra organismos patógenos. El principal propósito del SI es reconocer a todas las células dentro del cuerpo y categorizarlas con el objeto de inducir un mecanismo de defensa apropiado. EL SI aprende a través de la evolución para poder distinguir entre los antígenos externos peligrosos (por ejemplo, bacterias, virus, etc.) de las células propias del cuerpo.

Los SI's tienen varias propiedades atractivas desde el punto de vista computacional. Es dificultoso encontrar otro sistema biológico que reúna tantas características poderosas y diversas. Estas son: Reconocimiento de patrones, unicidad, auto-identificación, diversidad, ninguna célula es indispensable para el funcionamiento del sistema inmune, autonomía, múltiples capas de diferentes mecanismos actúan en forma cooperativa, detección de anomalías, tolerancia a las fallas, sistema distribuido, robustez, aprendizaje y memoria, auto-organización e integración con otros sistemas.

Estas características junto con un buen conocimiento de cómo trabaja el sistema inmune son excelentes motivaciones para desarrollar un Sistema Inmune Artificial (SIA) para el manejo de restricciones. Además este tipo de heurística bio-inspirada no ha sido ampliamente validada en el contexto de optimización de problemas restringidos.

Los modelos basados en los principios del SI, tal como la Teoría de la Selección Clonal [4, 1, 12], el Modelo *Immune Network* [15, 8] o el Algoritmo de Selección Negativa [14] han incrementado

sus aplicaciones en el campo de la ciencia e ingeniería [10], tal como: seguridad, detección de virus, monitoreo de procesos, reconocimiento de patrones, optimización numérica, etc.

Es por ello que la heurística Artificial Immune System (AIS) ha comenzado a ser estudiada por parte de los integrantes de la línea. Cuando este modelo computacional AIS es comparado con otros bien establecidos modelos de la inteligencia computacional como los algoritmos evolutivos, las colonias de hormigas, las redes neuronales, entre otras, puede decirse que la AIS está aún en los primeros años de su infancia .

Con respecto a la aplicación de AIS en optimización de funciones con restricciones, poco es el trabajo reportado hasta el momento [31, 32, 16, 7, 2, 13, 19, 9].

## 2.1. Trabajo Actual y Futuro

Actualmente el estudio está centrado en el desarrollo de uno o varios operadores de mutación (principal motor de búsqueda de los AIS) capaz de explorar eficientemente el espacio de búsqueda y a la vez explotar eficazmente las soluciones factibles encontradas. Todos los algoritmos desarrollados hasta el momento están basados en el algoritmo CLONALG propuesto por Nunes y Von Zuben [20, 21] y en un estudio posterior, el algoritmo CLONALG *with controlled and uniform mutations* propuesto por Cruz Cortés, Trejo-Pérez y Coello Coello [9]. Inicialmente, CLONALG fue utilizado para resolver problemas de reconocimiento de patrones y optimización multimodal. Posteriormente, en [9] CLONALG fue extendido para trabajar con problemas de optimización con restricciones.

En general el mecanismo de trabajo de CLONALG es el siguiente: dada una población de anticuerpos, donde cada uno de ellos tiene asociado un valor de afinidad (correspondiente al valor de la función objetivo), se seleccionan para ser clonados aquellos individuos con mejor afinidad. Los anticuerpos seleccionados serán clonados proporcionalmente a su afinidad generando un repertorio de clones. Cada uno de los clones es hipermutado en forma inversamente proporcional a su afinidad. Para luego seleccionar los mejores individuos entre la población de clones y la población de anticuerpos. Por último se reemplazan los individuos de menor afinidad por individuos generados aleatoriamente.

Algunas consideraciones deben ser tomadas cuando se aplica CLONALG a problemas con restricciones, ellas son: determinar la afinidad de cada anticuerpo involucra determinar la factibilidad del anticuerpo, cantidad de restricciones que satisface, el valor de la función objetivo para dicho anticuerpo y el grado de violación de las restricciones.

El primer operador de mutación desarrollado trabaja de la siguiente forma: a cada variable de decisión se le suma o resta, con una probabilidad del 80 %, un valor aleatorio uniformemente distribuido entre 0 y 1 multiplicado por un valor aleatorio uniformemente distribuido en el rango de dicha variable dividido por el número de generación actual o con una probabilidad del 20 % a cada variable de decisión se le suma o resta un valor aleatorio uniformemente distribuido entre 0 y 1 multiplicado por un valor aleatorio uniformemente distribuido en el rango de dicha variable dividido por el número de restricciones del problema. La idea general del primer tipo de cambio es disminuir el incremento en la variación de cada variable de decisión a medida que avanza la búsqueda. La idea del segundo tipo de cambio es incorporar cambios en las variables de decisión teniendo en cuenta las restricciones del problema. Si el problema tiene pocas restricciones entonces los cambios serán más drásticos, ahora, si el problema posee muchas restricciones entonces la búsqueda actuará en forma local.

Este operador fue validado con 13 funciones de prueba de la literatura especializada [27], logrando obtener buenos resultados.

Posteriormente, se desarrolló otro operador de mutación que permitió superar los resultados obtenidos anteriormente. Aquí, el operador de mutación realiza una distinción entre las soluciones factibles y las no factibles. Si una solución es factible entonces sólo una posición de la cadena que representa al anticuerpo se cambia por un valor aleatorio uniformemente distribuido en el rango de dicha variable.

Si la solución es no factible entonces, con una probabilidad del 50 % a cada variable de decisión se le suma o resta un valor aleatorio uniformemente distribuido entre 0 y 1 multiplicado por un valor aleatorio uniformemente distribuido en el rango de dicha variable dividido por el número de generación actual y al resultado de esta división se lo multiplica por la cantidad de clones, o bien a cada variable de decisión se le suma o resta un valor aleatorio uniformemente distribuido entre 0 y 1 multiplicado por un valor aleatorio uniformemente distribuido en el rango de dicha variable dividido por la multiplicación de el número de generación actual y la cantidad de clones.

Este operador también fue validado con 13 funciones de prueba de la literatura especializada [27].

Los resultados obtenidos hasta el momento han sido reportados y en este momento el artículo se encuentra en proceso de evaluación. El trabajo actual consiste en la implementación de otros modelos de sistemas inmunes artificiales.

### 3. Particle Swarm Optimizer

La metaheurística de propósito general Particle Swarm Optimizer (PSO)[30] está basada en la metáfora de cómo algunas especies comparten información y luego la utilizan para moverse a los lugares donde se encuentra el alimento que necesitan.

PSO cuenta con una población de entidades llamadas “partículas”, la cual es inicializada y luego, en base a una función de evaluación (función objetivo), se determina cuán bueno es cada individuo. En un proceso iterativo, la heurística afina gradualmente la dirección que las partículas deben seguir de forma de encontrar la zona del espacio de búsqueda donde se encuentran las buenas soluciones. Para este ajuste de direcciones se tiene en cuenta la propia experiencia de cada partícula y la experiencia de sus vecinos. Dependiendo del tamaño de vecindario que se considere existen dos modelos básicos de PSO: El algoritmo PSO global (*gbest*) que considera como vecindario de cada partícula a la totalidad del swarm y el algoritmo PSO local (*lbest*) que permite definir distintas topologías de vecindarios de menor tamaño para cada partícula del swarm.

#### 3.1. Trabajo Actual y Futuro

Se comenzó a trabajar con una versión básica del algoritmo PSO global, a la cual se le incorporó un esquema sencillo de manejo de restricciones y se incluyó un operador de mutación uniforme. A esta versión se le dio el nombre de CPSO. Para evaluar los resultados de los algoritmos se seleccionó un conjunto de 13 funciones de prueba [26] que tienen características particulares y diferentes entre sí. Los resultados iniciales obtenidos con CPSO no fueron los esperados dado que no eran competitivos con respecto a los reportados por otros algoritmos presentados en la bibliografía especializada.

Posteriormente, se utilizó el modelo de PSO local, incorporando a CPSO vecindarios lógicos y se cambió el operador de mutación uniforme por uno dinámico a fin de evitar la dispersión excesiva de las partículas dentro del espacio de búsqueda.

Aunque los resultados habían mejorado, para la mayoría de funciones de prueba las mejores soluciones encontradas no se acercaban a los óptimos conocidos. En este punto de la evaluación de CPSO se observó que la diversidad de la población mantenida durante la ejecución del algoritmo era alta, lo que producía la pérdida de buenas soluciones. Este inconveniente se solucionó incorporando una nueva fórmula de actualización de las partículas, usando la idea presentada por Kennedy en *Gaussian Bare Bones PSO* [17]. Esta fórmula elige una nueva posición aleatoria para las partículas usando una distribución Gaussiana. CPSO, entonces, elige con una probabilidad del 50 % esta fórmula o la fórmula típica de actualización de las partículas.

Los resultados alcanzados con esta última versión de CPSO fueron muy competitivos, aún comparados con el algoritmo Stochastic Ranking [26]. Este último es el algoritmo que ha reportado los

mejores resultados alcanzado para cada uno de los 13 problemas. También se compararon los resultados con una versión de PSO presentada por Toscano Pulido y Coello Coello [25] adaptado para resolver el conjunto de las 13 funciones evidenciándose, para algunas de las funciones de prueba, la superioridad de CPSO. Todos los resultados anteriormente descriptos fueron enviados a un congreso internacional para su evaluación.

Al momento se está estudiando la posibilidad de incorporar nuevas distribuciones de probabilidades para guiar la dirección de las partículas.

Adicionalmente se está trabajando en la adaptación de la heurística PSO para problemas de optimización multicriterio, en primer instancia, para ambientes estáticos y, posteriormente, si los resultados obtenidos son buenos se extenderá para su aplicación en ambientes dinámicos.

## Referencias

- [1] G. L. Ada and S. G. Nosal. The clonal selection theory. *Scientific American*, 257(2):50-57, 1987.
- [2] Jerzy Balicki. Multi-criterion Evolutionary Algorithm with Model of the Immune System to Handle Constraints for Task Assignments. In Leszek Rutkowski, Jörg H. Siekmann, Ryszard Tadeusiewicz, and Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004, 7th International Conference. Proceedings*, pages 394–399, Zakopane, Poland, June 2004. Springer. Lecture Notes in Computer Science. Volume 3070.
- [3] G. Bilchev and I. C. Parmee. Constrained and multi-modal optimization with ant colony search model. In *Proceeding of 2<sup>nd</sup> International Conference on Adaptive Computing in Engineering Design and Control*, University of Plymouth, UK, 1996.
- [4] F. M. Burnet. Immunological recognition of self. Nobel Lecture, pages 6-12, 1959.
- [5] C. A. Coello Coello. Constrain handling using evolutionary multiobjective optimization technique. In *Engineering Optimization*, volume 3, pages 275–308, 2000.
- [6] C. A. Coello Coello and E. Mezura Montes. Handling constraints in ga using dominance based tournaments. In *Proceeding of 5<sup>th</sup> International Conference on Adaptive Computing in Engineering Design and Manufacture*, volume 5, pages 273–284, University of Exeter, UK, 2002. Springer-Verlag.
- [7] Carlos A. Coello Coello and Nareli Cruz-Cortés. Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 36(5):607–634, October 2004.
- [8] A. Coutinho. The network theory: 21 years later. *Scandinavian Journal of Immunology*, 42:3-8, 1995.
- [9] Nareli Cruz Cortés, Daniel Trejo-Pérez, and Carlos A. Coello Coello. Handling constrained in global optimization using artificial immune system. In Christian Jacob, Marcin L. Pilat, Peter J. Bentley, and Jonathan Timmis, editors, *Artificial Immune Systems. 4th International Conference, ICARIS 2005*, pages 234–247. Springer. Lecture Notes in Computer Science Vol. 3627, Banff, Canada, August 2005.
- [10] D. DasGupta. Artificial immune systems and their applications. springer-Verlag New York, Inc, 1998.

- [11] K. Deb. An efficient constrain handling method for genetic algorithms. In *Computer Methods in Applied Mechanics and Engineering*, volume 186, pages 311–338, 2000.
- [12] D. R. Forsdyke. The origins of the clonal selection theory of immunity as a case study for evaluation in science. *The FASEB Journal*, 9:164-166, 1995.
- [13] H. Chueh G. C. Luh and W. W. Liu. MOIA: Multi Objective Immune Algorithm. *Engineering Optimization*, 35(2):143–164, 2003.
- [14] N. K. Jerne. The natural selection theory of antibody formation. *Proceeding of the National Academy of Sciences USA*, 41:849-856, 1955.
- [15] N. K. Jerne. Towards a network theory of the imune system. *Ann. Immunol. (Inst. Pasteur)*, 125(C):373-389, 1974.
- [16] Johnny Kelsey and Jon Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian F. Miller, editors, *GECCO*, volume 2723 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2003.
- [17] J. Kennedy. Bare bones particle swarms. In *IEEE Swarm Intelligence Symposium*, pages 80–87, 2003.
- [18] G. Leguizamón and Z. Michalewicz. A new version of ant system for subset problems. In *Proceeding of 1999 Congress on Evolution Computation*, pages 1459–1464, IEEE Press, Piscataway, NJ, 1999.
- [19] G. C. Luh and H. Chueh. Multi-objective optimal design of truss structure with immune algorithm. *Computers and Structures*, 82:829–844, 2004.
- [20] L. Nunes de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 669–674, Honolulu, Hawaii, May 2002.
- [21] L. Nunes de Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, 2002.
- [22] Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.
- [23] J. Pannier and J. Kao Hao. Simulated annealing and tabu search for constraint solving. In *Artificial Intelligence and Mathematics V, Electronic proceedings*, 1998. <http://rutior.rutgers.edu/anai/proceeding.html>.
- [24] M. Protasi and R. Battiti. Reactive local search technique for the maximum conjunctive constrain satisfaction problem. In *Discrete Applied Mathematics*, pages 3–27, 1999.
- [25] G. Toscano Pulido and C. A. Coello Coello. A constrained-handling mechanism for particle swarm optimization. In *Congress on Evolutionary Computation*, pages 1396–1403, Portland, Oregon, USA, 2004.

- [26] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. In *IEEE Transactions on Evolutionary Computation*, volume 3, pages 284–294, 2000.
- [27] Thomas P. Runarsson and Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.
- [28] H. Schoofs and B. Naudts. Ant colonies are good at solving constrain satisfaction problems. In *Proceeding of 2000 Congress on Evolution Computation*, volume 2, pages 1190–1195, IEEE Service Center, Piscataway, NJ, 2000.
- [29] M. Schütz, G. Leguizamon, and Z. Michalewicz. An ant colony system for the maximum independent set problem. In *Proceeding of 2001 Argentinian Congress of Computer Science*, pages 1027–1040, El Calafate, Argentina, 2001.
- [30] Russell Eberhart y James Kennedy. A new optimizer using particle swarm theory. In *Proceeding of the Sixth International Symposium on Micromachine and Human Science*, page 39, 1995. Nagoya, Japan.
- [31] J. Yoo and P. Hajela. Enhanced GA Based Search Through Immune System Modeling. In *3rd World Congress on Structural and Multidisciplinary Optimization*, Niagara Falls, New York, May 1999.
- [32] J. Yoo and P. Hajela. Immune network modelling in design optimization. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 167–183. McGraw-Hill, London, 1999.